<u>**Amendments to the Claims:**</u>

This listing of claims will replace all prior versions, and listings, of claims in the application:

<u>**Listing of Claims:**</u>

1.(Canceled) 1.   System for exchange of data between different clients (2,4,6) by using a central synchronization server (10) having a connection to said clients (2,4,6) and a connection to a Back End data store (24,26), wherein said clients having a program for creation of data to be synchronized, and a Sync Engine (12) for performing synchronization with said central synchronization server (10), wherein said system is characterized by the further components:

a single Back End neutral interface (CAF- interface; 22)) with said Sync Engine (12), and

at least one component (content adapter; 28,30) comprising a Back End dependent part having an interface with said single Back End neutral interface (22) and said assigned Back End data store (24, 26).

2. (currently amended) A system <s>System</s> according to claim 22 <s>1</s>, wherein <u>each of</u> said components<u>s</u> <s>(28,30)</s> further comprises an abstract Back End independent part, wherein said abstract Back End independent part provides common

2

functionalities for use by all the Back End dependent parts.

3. (currently amended) A system ~~System~~ according to claim 2, wherein <u>each of said at least one back end</u> ~~Back End~~ data store ~~type (24,26)~~ is assigned ~~an~~ <u>its</u> own <u>said</u> component ~~(28,30)~~.

4. (currently amended) A system ~~Server~~ according to claim <u>22</u> ~~1~~, wherein said exchange of data is synchronization of data.

5. (currently amended) A system ~~System~~ according to claim 2, further <u>comprising</u> ~~comprises~~ a cache ~~(50)~~ for permanently buffering of updates of said <u>at least one back end</u> ~~Backend~~ data store ~~(24)~~ and said <u>clients</u> ~~Clients~~, and <u>each</u> said component ~~(28,30)~~ comprises a caching mechanism for controlling and executing buffering updates into said cache and replicating buffered updates to said respective clients and said <u>assigned back end</u> ~~Backend~~ data store ~~(24)~~.

6. (currently amended) A system ~~System~~ according to claim 5, wherein said caching mechanism ~~having~~ <u>has</u> a Back End Monitor ~~(60)~~.

7. (currently amended) A system ~~System~~ according to claim 5, wherein said caching mechanism ~~further having~~ <u>includes</u> a Cache Monitor ~~(70)~~.

8. (currently amended) A system ~~System~~ according to claim 6, wherein said caching mechanism further ~~having~~ <u>includes</u> a Back End Manager ~~(80)~~.

3

9. (currently amended) A system ~~System~~ according to claim 6, wherein said caching mechanism provides for each <u>of said at least one back end</u> ~~Back End~~ data store ~~type (24) an~~ <u>its</u> own Back End Monitor, Cache Monitor, and Back End Manager with its Back End dependent <u>part</u> ~~(60′,70′,80′)~~ and its abstract Back End independent part ~~(60′′,70′′,80′′)~~.

10. (currently amended) A system ~~System~~ according to claim 5, wherein said caching mechanism further comprises a persistent store ~~(40)~~.

11. (currently amended) A system ~~System~~ according to claim 7, <u>wherein</u> said Cache Monitor ~~(70)~~ replicates updates from said cache to ~~the Back End~~ <u>the associated one of said at least one back end</u> data store ~~(24)~~ in a batch or a continuous trickle mode.

12. (currently amended) A system ~~System~~ according to claim 6, wherein said Back End Monitor ~~(60)~~ replicates updates between said cache ~~(50)~~ and the <u>associated one of said at least one back end</u> ~~Backend~~ data store ~~(24)~~ in a batch or a continuous trickle mode.

13. (currently amended) A system ~~Systems~~ according to claim 5, wherein said cache ~~(50)~~ and said <u>at least one back end</u> ~~Backend~~ data store ~~(24)~~ are databases.

14. (currently amended) A system ~~System~~ according to claim <u>22</u> ~~1~~, wherein said clients are mobile clients.

15. (currently amended) A system ~~System~~ according to claim

4

4, wherein ~~the~~ <u>SyncML is employed as a</u> synchronization protocol ~~is SyncML~~.

16. (cancelled) Method for synchronization of data by using a system according to claim 1-15, comprising the steps of:

Receiving sync session request from said Client,

Authenticating said Client against said Sync Server,

Receiving update from said client,

Authenticating said client against Back End data store via said CAF interface using Back End Monitor,

Creating of data objects and filling in said update received from said client by said Sync Server,

Calling said CAF interface and handing over said data objects,

Selecting the appropriate Back End specific part of said component assigned to said Back End data store,

Transforming said data objects CAF into a Back End specific format,

Executing said updates by calling Back End specific part and passing the data objects to it.

17. (currently amended) <u>A method</u> ~~Method~~ according to claim

5

23 ~~16~~, wherein the back end ~~said Back End~~ specific part is inherited from ~~said abstract Back End~~ an abstract back end independent part assigned to said ~~Back End~~ back end data store.

18. (canceled) Method according to claim 16, wherein said data objects are used to pass said client request to Back End specific parts.

19. (currently amended) A method ~~Method~~ according to claim 23 ~~18~~, wherein said data objects contain meta data.

20. (currently amended) A method ~~Method~~ according to claim 23 ~~16~~, wherein a synchronization protocol used exclusively between said client ~~clients~~ and said ~~central synchronization~~ synch server is SyncML and ~~said updates~~ the update received by said ~~central synchronization~~ synch server ~~are~~ is presented as XML documents.

21. (canceled) Computer program product stored in the internal memory of a digital computer, containing parts of software code to execute the method in accordance with claims 16 to 20 if the product is run on the computer.

22. (new) A system for exchange of data between a plurality of clients and at least one back end data store by using a central synchronization server having a connection to said, said clients generating data to be synchronized, said system comprising:

a sync engine for performing synchronization with said central synchronization server and connected to said

central synchonization server;

a single back end neutral interface associated with and connected to said sync engine; and

a component assigned to each of said at least one back end data store, each of said components comprising a back end dependent part having an interface with said single back end neutral interface and an interface with said assigned back end data store.

23. (new) A method for synchronization of data, said method comprising the steps of:

receiving a sync session request from a client;

authenticating said client against a sync server;

receiving an update from said client;

authenticating said client against a back end data store via a content adaptable framework interface using a back end monitor;

creating data objects and filling in the update received from said client by said sync server;

calling said content adaptable framework interface and forwarding said data objects;

selecting an appropriate back end specific part of a component assigned to said back end data store;

transforming a content adaptable framework of said data objects into a back end specific format; and

executing the update by calling the back end specific part and passing the data objects to the back end specific part.